Cimcor Cryptographic Module v1.0

FIPS 140-2 Level 1 Security Policy

Version: 1.6
Last Updated: February 2, 2010

CIMCOR, Inc.
Corporate Headquarters
8252 Virginia Street, Suite C
Merrillville, IN 46410
Toll Free: 877-424-6267
Intl: (219) 736-4400
Fax: (219) 736-4401

Revision History

Authors	Date	Version	Comment
Cimcor, Inc.	2009-02-08	1.0	Initial draft
Cimcor, Inc.	2009-06-20	1.1	New label for block diagram in section 1. Two Certificates columns for Algorithms table in section 5, removed D-H. References to both level 1 and level 2 as appropriate.
Cimcor, Inc.	2009-06-24	1.2	2.0: designate platforms 2.1: specify missing platforms 4.1: renumber 5.0: drop superfluous column
Cimcor, Inc.	2009-06-26	1.3	3.1: remove unnecessary reference to PINs and passwords, refer to D-H primitives.
Cimcor, Inc.	2009-07-05	1.4	New block diagram
Cimcor, Inc.	2009-07-27	1.5	Revised for Level 1 targets
Cimcor, Inc.	2010-02-02	1.6	Updated to incorporate comments

Table of Contents

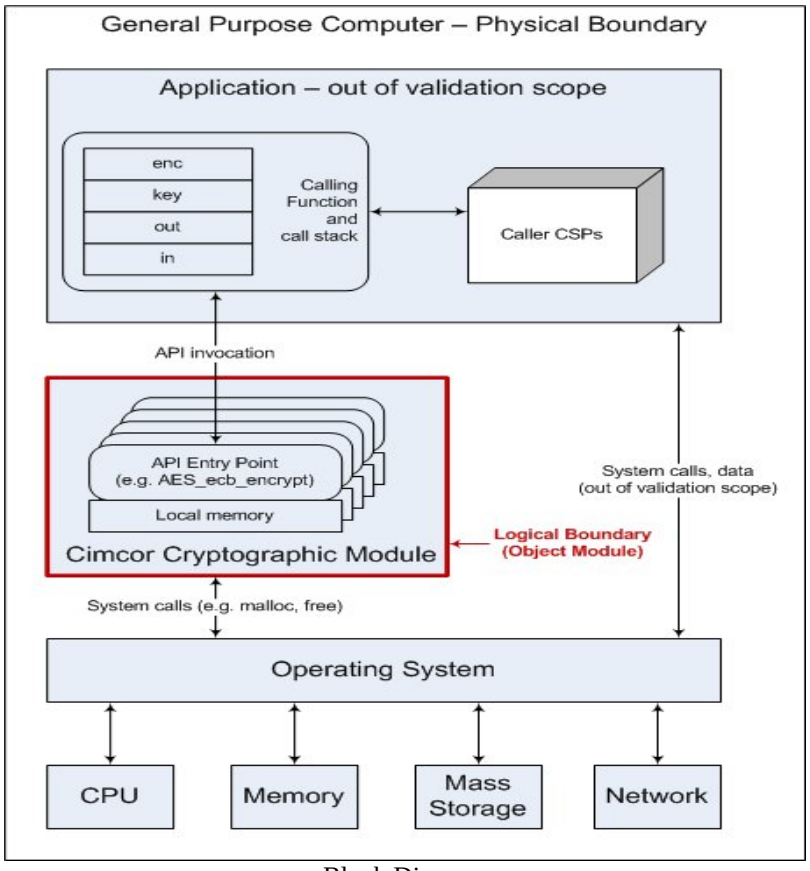
1. Introduction

This document comprises the non-proprietary FIPS 140-2 Security Policy for the Cimcor Cryptographic Module (Software Version 1.0).

FIPS 140-2, *Security Requirements for Cryptographic Modules*, specifies the requirements for cryptographic modules. For more information about the FIPS 140-2 standard and the cryptographic module validation process see http://csrc.nist.gov/cryptval/.

2. Cimcor Cryptographic Module Overview

The Cimcor Cryptographic Module (Module) is a software library that provides symmetric and asymmetric encryption, hashing, and random number generation functions on a wide variety of computing platforms. The Module is the specific shared library file with the name given in the *Supported Platforms* table in section 2.1 below.



Block Diagram

For FIPS 140-2 validation purposes the Module is a multi-chip embedded module. The logical cryptographic boundary of the Module is the shared library file. The Module exchanges data only with the calling application, and does not perform any network or inter-process communication and does not read or write any persistent storage.

The Module relies on the host operating system for authentication of operators and protecting and clearing authentication data when the computer is powered down or restarted. As FIPS 140-2 Level 1 does not require authentication, and no claim is made beyond Level 1, the authentication function performed by the operational environment is not covered further.

Requirement	Compliance Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference/Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Overall	1

Compliance

2.1 Supported Platforms

The following operating system platforms were used to operationally test and validate the Module to FIPS 140-2 requirements. These tested platforms provide the highest level of assurance that the module will operate correctly.

Operating System configurations:

Operating System	Hardware Platform	Module File Name
Microsoft Windows Vista	Dell Optiplex GX620	libosslfips.dll
Microsoft Windows Server 2008	Dell Optiplex GX620	libosslfips.dll

Supported Platforms

2.2 Ports and Interfaces

The physical ports of the Module are the same as the General Purpose Computer (GPC) on which it is executing. The logical interface is a C-language application program interface (API).

The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API functions. The Status Output interface includes the return values of the API functions. In the error state data output is disabled.

Interface	*Physical Port*	*Module Interface*
Data Input	Physical Ports of a GPC	API input parameters
Data Output	Physical Ports of a GPC	API output parameters
Control Input	Physical Ports of a GPC	API function calls and parameters, other than those used by Data Input and Output interfaces
Status Output	Physical Ports of a GPC	API return codes
Power Input	Power Port of a GPC	N/A

Ports and Interfaces

3. Roles, Services, and Authentication

3.1 Roles and Services

The Module meets all FIPS 140-2 Level 1 requirements for Roles and Services, implementing both User and Crypto-Officer roles. At Level 1, authentication is not required.

The User and Crypto-Officer roles are implicitly assumed by the entity accessing services implemented by the Module. The Crypto-Officer role is implicitly entered when installing the Module or performing system administration functions on the host operating system. The Module does not implement any security relevant functions exclusive to the Crypto-Officer role, and the Crypto-Officer may access all Module services available to the User role. The allocation of functions to roles is fixed and cannot be changed by any runtime or configuration parameters.

Service	***Role***	***CSP***	***Access***
Symmetric encryption/decryption	User, Crypto-Officer	AES and TDES symmetric keys	read/write/execute volatile memory
Key wrapping for key transport	User, Crypto-Officer	RSA public/private key pairs	read/write/execute volatile memory
Key agreement primitives	User, Crypto-Officer	Diffie-Hellman keys	read/write/execute volatile memory
Digital signature	User, Crypto-Officer	RSA and DSA asymmetric keys	read/write/execute volatile memory
Symmetric key generation	User, Crypto-Officer	AES, TDES and HMAC symmetric keys	read/write/execute volatile memory
Asymmetric key generation	User, Crypto-Officer	RSA, DSA and Diffie-Hellman keys	read/write/execute volatile memory
Keyed Hash (HMAC)	User, Crypto-Officer	HMAC keys	read/write/execute volatile memory
Message digest (SHS)	User, Crypto-Officer	none	read/write/execute volatile memory
Random number generation (ANSI X9.31)	User, Crypto-Officer	RNG seed and seed key	read/write/execute volatile memory
Show status	User, Crypto-Officer	none	execute volatile memory
Module initialization	User, Crypto-Officer	none	execute volatile memory
Self-test	User, Crypto-Officer	Integrity-check HMAC key	execute volatile memory
Zeroize	User, Crypto-Officer	all symmetric and asymmetric keys, as well as parameters other than those used by Data Input and Output Interfaces	write volatile memory

Roles and Services

The Module does not store any CSPs in persistent storage, and does not read or write any persistent storage. All CSPs are stored in volatile memory only.

Note that only the private key components of public/private key pairs are CSPs. The public keys are assumed to be publicly visible.

4. Secure Operation

The tested operating systems segregate user processes into separate process spaces. Each process space is an independent virtual memory area that is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the process that invokes it, and thus satisfies the FIPS 140-2 level 1 requirement for a single user mode of operation. The host operating system provides authentication services to prevent unauthorized access to the Module.

A complete revision history of the source code from which the Module was generated is maintained in a version control database.

Upon initialization of the Module by invoking the FIPS mode initialization API call, the module will run its power up self-tests. Specifically, the module is initialized into the FIPS mode by calling FIPS_mode_set(TRUE); a TRUE return value from this call indicates successful completion of power up self-tests and confirmation that the module is operating in the FIPS approved mode of operation.

The self-tests can be called on demand by restarting the module (i.e., reloading the module and re-invoking the FIPS mode initialization API call). The calling application can query the current status of the FIPS mode of operation at any time.

The Module implements the following non-FIPS Approved algorithms which are allowed for use in FIPS mode: RSA encrypt/decrypt, used for key wrapping; Diffie-Hellman primitives, used for key agreement.

In the non-FIPS Approved mode of operation (prior to successful completion of the power up self-tests or subsequent to explicitly exiting the FIPS mode of operation) the same algorithms are available as in the FIPS Approved mode of operation, with the addition of algorithms and key sizes as noted in the Non-Approved Algorithms table in Section 5 below.

Invalid input via API function calls will not compromise the security of the Module.

4.1 Installation

Installation consists of copying the shared library file to the appropriate location where it can be referenced by the host operating system at application runtime.

Installation instructions:

1. Copy the shared library file to the appropriate location on the host system for protected system libraries.
2. As appropriate define or register the shared library for reference by the operating system (O/S) run-time loader. This step will vary depending on the O/S and whether the shared library is to be installed for global access by all users or only for use by a specific application. For Microsoft Windows simply placing the shared library in the same directory as the calling application suffices for use by that application. .

4.2 Mitigation of Other Attacks

The Module does not implement security mechanisms beyond those required for FIPS 140-2 level 2 validated modules.

4.3 Physical Security

The Module is a software library and thus does not claim any physical security.

4.4 Security Rules

This section documents the security rules enforced by the cryptographic module in the general purpose computing environment to implement the security requirements of this FIPS 140-2 module.

The operating environment of the cryptographic module shall provide two distinct operator roles. These are the User role, and the Cryptographic Officer role.

When the operating environment of the cryptographic module has not been placed in a valid role, calling applications do not have access to any cryptographic services.

Calling applications can command the module to perform the power up self-test by invoking the FIPS mode initialization API call.

Power up self-tests do not require any action other than invoking the FIPS mode initialization API call.

Data output is inhibited during key generation, self-tests, zeroization, and error states.

Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module. CSPs are not stored beyond the lifetime of API function calls.

The module ensures that the seed and seed key inputs to the Approved DRNG are not equal.

There are no restrictions on which keys or CSPs are zeroized by the zeroization service.

The module does not support a maintenance interface or role.

The module does not support manual key entry.

The module is software only and does not have any external input/output devices.

The module does not enter or output plaintext CSPs except as passed by the calling application during the performance of a cryptographic operation.

The module does not output intermediate key values.

5. Cryptographic Key Management

The Module does not persistently store any Critical Security Parameters (CSPs). All CSP key input and key output operations are managed by the calling application using the Module API to transfer CSPs across the cryptographic module boundary.

CSP	Source	Usage
RSA Keys	Generated using module PRNG	Sign and verify
DSA Keys	Generated using module PRNG	Sign and verify
AES Keys	Generated using module PRNG	Encryption and decryption
TDES Keys	Generated using module PRNG	Encryption and decryption
HMAC Keys	Generated from external key and data	Message integrity
PRNG Keys	Seed, key	Key generation

CSPs

The Module supports the following FIPS approved algorithms:

Algorithm	Certificate	Notes
AES	1121	
TDES	818	
DSA	364	1
RNG ANSI X9.31	624	
RSA (X9.31, PCKS#1.5, PSS)	530	1
SHS (SHA-1, SHA-224/256/384/512)	1044	
HMAC (HMAC-SHA-1, HMAC-SHA- 224/256/384/512)	632	

Approved Algorithms

Note 1: Does not support a key size of less than 1024 bit when in the FIPS mode of operation.

The Module supports the following non-FIPS Approved algorithms:

Algorithm	Notes
RSA encrypt/decrypt	2
Diffie-Hellman primitives	3
Blowfish	4
Camellia	4
DES	4
Idea	4
RC2	4
RC4	4
RC5	4
MD2	4
MD4	4
MD5	4
Mdc2	4
Ripemd	4

Non-Approved Algorithms

Note 2: RSA (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength) allowed in FIPS mode.

Note 3: Diffie-Hellman primitives (key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength) allowed in FIPS mode.

Note 4: Available in non-FIPS mode only.

5.1 Key Zeroization

Keys residing in internally allocated data structures can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access.

Only the process that creates or imports keys can use or export them. No persistent storage of key data is performed by the Module. All API functions are executed by the invoking process in a non-overlapping sequence such that no two API functions will execute concurrently. The Module zeroizes dynamically allocated volatile memory when de-allocating that memory.

The Module provides a zeroization function for use by calling applications. Rebooting of the system will zeroize any keys present in volatile RAM.

5.2 Self-Tests

The Module performs both power up self-tests at module initialization and conditional tests during operation. Input, output, and cryptographic functions cannot be performed while the Module is in a self-test or error state as the module is single threaded and will not return to the calling application until the power up self-tests are complete. If the power up self-tests fail subsequent calls to the module will fail and thus no further cryptographic operations are possible.

Power Up Self-Tests

Algorithm	Test
AES	KAT
Triple-DES	KAT
DSA signatures	pairwise consistency test, sign/verify
RSA signatures	KAT
ANSI X9.31 PRNG	KAT
HMAC-SHA-1	KAT
HMAC-SHA-224	KAT
HMAC-SHA-256	KAT
HMAC-SHA-384	KAT
HMAC-SHA-512	KAT
SHA-1	KAT[1]
SHA-224	KAT[1]
SHA-256	KAT[1]
SHA-384	KAT[1]
SHA-512	KAT[1]
Module integrity check	HMAC-SHA-1

Power Up Self-Tests

1 Tested as part of the HMAC known answer tests.

Conditional Self-Tests

Algorithm	Test
DSA key pair generation	pairwise consistency
RSA key pair generation	pairwise consistency
PRNG	continuous RNG test

Conditional Self-Tests

A single API call is required to initialize the Module for operation in the FIPS 140-2 Approved mode. When so initialized the Module performs all security functions and cryptographic algorithms in FIPS Approved mode.

The initialization function verifies the integrity of the runtime executable using a HMAC-SHA-1 digest computed at build time. If this computed HMAC-SHA-1 digest matches the stored known digest then the power up self-test, consisting of the algorithm specific Pairwise Consistency and Known Answer tests, is performed. If any component of the power up self-test fails subsequent invocation of any cryptographic function calls is disabled. Any such failure is a hard error that can only be recovered by reloading the Module.